











CURSO: Modulo 2. Desarrollo de aplicaciones web con Python

Práctica de laboratorio 6: Plantillas Jinja2

Objetivos

- 1. Identificar delimitadores de Jinja2
- 2. Utilizar plantillas de Jinja2 en aplicaciones para rendeerizar paginas HTML

¿Qué es Jinja2?

Jinja2 es un lenguaje de plantillas rico en características ampliamente utilizado en el ecosistema de Python. Se puede usar directamente en sus programas de Python y muchas aplicaciones más grandes lo usan como su motor de renderizado de plantillas.

Los lenguajes de plantilla permiten la creación de documentos basados en texto donde parte del contenido puede generarse dinámicamente. Los archivos resultantes pueden ser HTML, JSON, XML o cualquier cosa que use texto puro como codificación. La idea es capturar la lógica de negocios en el código mientras brinda herramientas de diseño de plantillas para controlar el flujo y el diseño del documento final.

¿Dónde se usa?

Algunos ejemplos notables de aplicaciones que usan Jinja2 son Ansible, Django, Flask, Salt y Trac. Muchos otros frameworks web de Python también lo usan, así como muchos otros proyectos de Python.

¿Qué tiene de bueno?

Jinja2 viene con muchas funciones excelentes:

- Estructuras de control (bucles y sentencias condicionales)
- Amplio conjunto de filtros y pruebas integrados
- Herencia de plantilla
- Soporte para filtros personalizados
- Escape de HTML
- Entorno de espacio aislado para renderizar de forma segura plantillas que no son de confianza
- Fácil de depurar
- Sintaxis configurable

¿Como funciona?





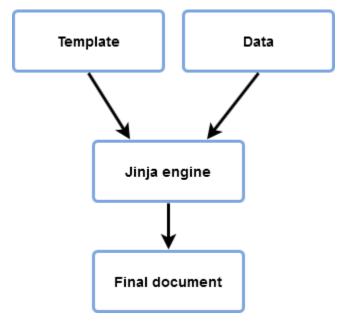








Jinja2 esencialmente necesita dos ingredientes de origen, plantilla y datos que se utilizarán para representar el documento final.



Delimitadores Jinja

Jinja usa varios delimitadores en las cadenas de plantilla.

- {% %} Sentencias
- {{ }} variables y expresiones para imprimir en la salida de la plantilla
- {# #} comentarios que no están incluidos en la salida de la plantilla

Ejercicios

Cree una nueva aplicación Django con lo siguiente:

Ejercicio 1

- Una plantilla base.html para que otros la hereden
- Una ruta para /persona/<nombre>/<edad> que genera una plantilla que muestra el nombre y la edad ingresados para la URL. Esa plantilla debe heredar de base.html

Ejercicio 2. Crear una calculadora

- Tener una ruta para /calculadora que genera una plantilla llamada calc.html
- en calc.html, cree un formulario que tenga dos entradas (una con el nombre de num1 y otra con el nombre de num2 para números y un campo de selección con el nombre de cálculo con opciones para "sumar", "restar", "multiplicar" y "dividir".













- Cuando se envía el formulario, debe realizar una solicitud a una ruta llamada /math
- En su archivo python, acepte los valores del formulario y, según lo que contenga la solicitud, responda con la suma, la diferencia, el producto o el cociente.

Ejercicio 3.

- Crear una lista con 20 titulares de noticias
- Esta aplicación debe constar de dos rutas, / y /resultados
- Cuando el servidor recibe una solicitud a /, debe presentar una página HTML con un formulario que solicite al usuario una palabra clave (por ejemplo, "venta de armas"). Enviar el formulario debe generar una solicitud a /resultados, con la entrada del formulario pasada en la cadena de consulta.
- Cuando el servidor recibe una solicitud a /resultados, debe hacer lo siguiente:
- Tome la palabra clave de la cadena de consulta;
- Realice una búsqueda en la lista con los titulares que incluyen la palabra
- Filtre los artículos cuyos titulares no coincidan con la palabra clave pasada desde la cadena de consulta:
- Representar una página HTML con artículos coincidentes en forma de enlaces en una lista HTML.

Ejercicio 4.

- Crear una lista con nombres de usuarios y su edad
- Crear aplicación Di con la ruta /lista-usuarios
- Pasar la lista a una plantilla llamada lista-usuarios.html
- Renderizar en la plantilla mediante una tabla HTML (cabeceras, cuerpo y pie) los usuarios de la lista.
- Si el usuario inicia con letra a colocar su nombre en negritas.

Ejercicio 5.

Crear aplicación Django para capturar los siguientes datos en la URL /













| ← → ୯ ☆ | localhost:5000/ |
|----------------|-----------------|
| Name MVL | |
| Physics 50 | |
| Chemistry 60 | |
| Maths 70 | |
| submit | |

Enviar los datos por método post a la ruta /result, la cual debe aceptar POST y GET, validar el método post e imprimir lo siguiente

